

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

PAT-NO: JP404131970A
DOCUMENT-IDENTIFIER: JP 04131970 A
TITLE: GATE LOGIC GENERATING METHOD
PUBN-DATE: May 6, 1992

INVENTOR-INFORMATION:

NAME	COUNTRY
MATSUO, KOJI	
OE, KIMIO	
AMANO, NOBUTAKA	
MOTOMURA, TETSURO	
FUKUNAGA, AKIRA	
NISHISAKA, TOSHIO	

ASSIGNEE-INFORMATION:

NAME	COUNTRY
HITACHI LTD	N/A
HITACHI MICRO COMPUT ENG LTD	N/A

APPL-NO: JP02251974
APPL-DATE: September 25, 1990

INT-CL (IPC): G06F015/60

ABSTRACT:

PURPOSE: To attain logic without the upper limit of maximum bit width by generating the gate logic of a logical macro by copying and connecting a pattern gate table that a functional unit to constitute the logical macro is defined by a binary tree gate table corresponding to bit width.

CONSTITUTION: The pattern developing rule of a logical macro is retrieved from a pattern developing rule data table 111 in a step 102 with a macro kind 416 of logical macro data 415 extracted in a step 100 as a key. In a step 103, by keying a pattern gate table name described on a pattern gate defining data table is retrieved from a step 102, a pattern gate table described by a binary tree gate table is copied and connected for a bit width portion and the binary tree gate table to define a local function corresponding to the bit width of the logical macro is generated. Thus, logic without the upper limit of maximum bit width can be generated.

COPYRIGHT: (C) 1992, JPO&Japio

⑫ 公開特許公報(A) 平4-131970

⑬ Int.Cl.⁵

識別記号

庁内整理番号

⑭ 公開 平成4年(1992)5月6日

G 06 F 15/60

3 6 0 K

7922-5L

審査請求 未請求 請求項の数 1 (全7頁)

⑮ 発明の名称 ゲート論理生成方法

⑯ 特 願 平2-251974

⑰ 出 願 平2(1990)9月25日

⑱ 発 明 者 松 尾 浩 次 東京都小平市上水本町5丁目22番1号 日立マイクロコンピュータエンジニアリング株式会社内

⑲ 発 明 者 大 江 公 夫 東京都小平市上水本町5丁目22番1号 日立マイクロコンピュータエンジニアリング株式会社内

⑳ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

㉑ 出 願 人 日立マイクロコンピュータエンジニアリング株式会社
東京都小平市上水本町5丁目22番1号

㉒ 代 理 人 弁理士 小川 勝男 外1名

最終頁に続く

明 細 書

1. 発明の名称

ゲート論理生成方法

2. 特許請求の範囲

1. 論理動作レベルで記述した論理をゲート論理にコンパイルし、ゲートレベルでシミュレーションを行うシステムにおいて、論理動作記述で使用される定数、パラメタ、関数と演算子からなる演算式の構文解析を行い、演算子あるいは関数を1つ論理マクロとしてその名前と入出力関係を表す入出力パラメタとそのパラメタに対応したデータの個数を表すビット幅からなる論理マクロデータを抽出し、当該論理マクロの上記名前をキーにして当該論理マクロのひな型展開ルールをひな形展開ルールデータテーブルから検索し、当該論理マクロのひな型展開ルールに従い、当該論理マクロを構成している機能単位を二分木ゲートテーブルで定義したひな型ゲートテーブルをビット幅に応じて複写し接続することによって当該論理マクロのゲート論理を

生成することを特徴とするゲート論理生成方法。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、論理検証で用いられるゲートレベルシミュレーション用のゲート論理を、論理動作レベルから生成するゲート論理生成方法に関する。

〔従来の技術〕

ゲート論理生成方法の一つに関数を含む演算式からその機能と接続関係、ビット幅を抽出し論理マクロ展開を行い論理回路を生成する演算式論理展開がある。

論理マクロ展開は、論理マクロ(演算式から抽出)のひな型を予め定義しておき、このひな型論理を用いて論理生成を行う方式である。論理マクロのひな型論理は一般に論理マクロの論理機能の種別を表すマクロ種と論理マクロの論理機能の大きさを表すビット幅と生成される論理回路のパワー区分やスピード区分等の物理的特性を表わす回路特性と生成される論理回路を構成するゲートセットを表わす回路系により決まる。そのため、論

理マクロ展開方式は、ひな型論理の定義数を最小にするために、マクロ種、ビット幅、回路特性、回路系の4つの要因を汎用的に扱うことが必要である。

論理マクロ展開方式に関してこれまでに、マクロ種とビット幅を汎用的に扱う方式が第23回デザインオートメーションカンファレンス予稿集第594頁から第600頁(1986年)(Proc. of 23rd DAC, p.p. 594-600, 1986)において記載されている。しかしこれは論理設計の自動化のための論理マクロ展開方式であり、ファンアウト本数他の実装条件を十分に考慮したものになっている。シミュレーションのためのビット幅の上限を設けない論理マクロ展開方式については、今のところ文献が見当たらない。
〔発明が解決しようとする課題〕

演算式論理展開では、論理マクロを演算式から抽出するため、設計者は論理機能の大きさを表わすビット幅を意識することなく設計を行なう。そのため、明示的にビット幅を指定する固定式論理

力関係を表す入出力パラメタとその論理機能の大きさを表すビット幅からなる論理マクロデータを抽出するステップ1と、当該論理マクロのマクロ名をキーにして当該マクロのひな型展開ルールをひな型展開ルールデータテーブルから検索するステップ2と、当該論理マクロのひな型展開ルールに従いビット幅に応じて当該論理マクロの最小機能単位を二分木ゲートテーブルで定義したひな型ゲートテーブルを複写し接続して当該論理マクロのゲートイメージの二分木テーブルを定義するステップ3と前ステップの二分木テーブルからゲート論理を生成するステップ4からなるようにしたものである。

〔作用〕

上記ステップ1は、演算式から論理マクロを自動的に抽出する。それにより、設計者が論理ゲートを意識することなく機能設計することを可能にする。また、上記ステップ2、3は、論理マクロで指定された機能ビット幅分ひな型ゲートテーブルを複写し必要とされる機能とビット幅を持つ論

理マクロとは異なり、ビット幅の上限を定めることは望ましくない。

従来の論理マクロ展開方式は、マクロ種毎に必要な最大ビット幅分のひな型論理を定義しておき、ひな型論理変形ルールを用いて当該論理マクロの指定ビット幅に応じてひな型論理を変形し、所望の論理回路を生成する方式である。そのため、この方式はひな型論理で用意した最大ビット幅を超えるビット幅を論理展開することはできないという問題があった。

本発明の目的は、演算式による論理動作記述をゲートレベルシミュレーション用のゲート論理ファイルに展開するにあたり、最大ビット幅の上限がない論理を生成する方法を提供することにある。
〔課題を解決するための手段〕

上記目的を達成するために、本発明は、論理動作レベルで記述した論理をゲート論理にコンパイルし、ゲートレベルでシミュレーションを行うシステムにおいて、演算式の構文解析を行い、論理マクロ名(演算子、関数名)と論理マクロに入出

理回路を生成する。それにより、ビット幅の上限を設けない論理マクロ展開を可能とする。

〔実施例〕

以下、本発明の一実施例を図面により詳細に説明する。第2図はゲート論理シミュレーションの処理の流れである。まず既存論理からゲート論理ファイルを作成するまでを説明し、次に設計対象論理からゲート論理ファイルを作成するまでを説明し、最後にこれら二つのゲート論理ファイルから論理シミュレーションを行うまでを説明する。

既存論理は論理動作レベルで記述(210)し、論理入力ステップ200を経て論理動作ファイル211が作成される。論理動作ファイル211は、論理動作記述コンパイラステップ201によりゲート論理ファイル212に展開される。本発明は論理動作記述コンパイラを構成する重要な1つの機能である演算子論理展開ステップ202を実現する。

設計対象論理はゲートレベルで記述(213)を行い、論理入力ステップ203を経てゲート論

理ファイル214が作成される。

以上2つのゲート論理ファイル212, 214を入力し、それぞれのファイルに格納されている回路データを合体させて、論理シミュレーション204を実行する。

第3図に論理動作記述210の例を示す。機能ブロック300は、当該ブロックのブロック種を表すブロック識別子301、入出力関係を表す入力パラメタ302及び出力パラメタ303、論理動作を表す演算式304、ブロックの個別名称を表すブロック名305、他のブロックとの接続関係を表す信号線306からなる。この例では、入力パラメタ302のA, Bは4ビット幅であり、出力パラメタ303のZは1ビット幅であり、演算式304はAとBの'<='比較演算結果をZに転送することを表している。

論理動作記述コンパイラ201は、テキスト形式の論理動作ファイル211を入力しゲートレベル論理ファイル212を出力する。論理動作ファイルのうち演算式の部分をゲートレベル論理フ

行ない処理しやすい形に並べ替える(414)。論理マクロデータ抽出402は、並べ替えた演算式414を入力し演算子から論理マクロの論理機能の種別を表すマクロ種416を、入力パラメタと代入先のパラメタのビット幅から論理マクロの論理機能の大きさを表すビット幅417を抽出し、論理マクロの入出力関係を表す入力パラメタ418と代入先パラメタ419と共にテーブルにセットし論理マクロデータ415を抽出する。この例では、マクロ種416は'<='比較演算を、ビット幅417は4ビット幅を、入力パラメタ418はA, Bパラメタを、代入先パラメタ419をZパラメタを意味する。

ステップ101:本ステップは、ステップ100で抽出された論理マクロデータと予め用意されたひな型展開ルール、ひな型ゲート定義データを基に、二分木ゲートテーブル形式で表現された機能論理を生成するひな型論理展開処理を行なう。本ステップはひな型展開ルール検索102とひな型ゲート展開103の2ステップで構成される。以

イル212に展開するのが演算式論理展開202である。

第1図は本発明に基づく演算式論理展開202の処理のフローチャートである。この図においてステップ101が本発明に関する部分である。この図に基づき、演算式論理展開処理の処理手順を順次説明する。

ステップ100:本ステップは演算式ファイル110から演算式を読み込み構文解析後論理マクロデータを抽出する。

第4図に演算式、論理マクロ抽出ステップ100の処理例を示す。演算式410は、パラメタ411, 演算子412, 代入指示子413からなる。この例では、パラメタ411は複数ビットのA(0-3), B(0-3)と1ビットのZに分けられる。演算子412の'<='は関係演算子を意味する。代入指示子' :'は左辺から右辺への代入を意味する。字句解析, 構文解析401は字句解析で演算式を意味をなす最小単位(パラメタ, 演算子など)に切り離し、構文解析で演算式の構文チェックを

下、ステップ102及び103の詳細を説明する。

ステップ102:本ステップはステップ100で抽出した論理マクロデータ415のマクロ種416をキーにして、当該論理マクロのひな型展開ルールをひな型展開ルールデータテーブル111から検索する。

ステップ103:本ステップはステップ102で検索したひな型展開ルールデータに記述されたひな型ゲートテーブル名をキーに、ひな型ゲート定義データテーブルを検索し二分木ゲートテーブルで記述された当該ひな型ゲートテーブルをビット幅分だけ複写接続し、当該論理マクロのビット幅に応じた論理機能を定義する二分木ゲートテーブルを生成する。

第5図はひな型展開ルールデータテーブル111内に定義されているマクロ種'<='(比較演算)のひな型展開ルールの例である。ひな型展開ルール500はC言語で記述された関数と似た構造を持ち、論理マクロのマクロ種と対応するひな型展開ルール名501, ビット幅の指定と入出力パラ

メタの対応をとる仮引数502, 仮引数502の属性を宣言する仮引数宣言部503, ひな型展開ルール内部の接続関係を表す内部パラメタと制御文で使用する内部変数を宣言する内部宣言部504, ひな型ゲートテーブル展開文506, 508と制御文507, 509と接続関数文510からなる展開ルール記述部505で構成される。

第6図はひな型ゲート定義データテーブル112内に定義されているひな型ゲート種 ' \leq head' と ' halfsub ' のひな型ゲートテーブル例である。ひな型ゲートテーブル600, 601はひな型展開ルール501に記述されたひな型ゲート展開文506に対応するひな型ゲート種602と、ひな型ゲート展開文の引数パラメタとひな型ゲートテーブルの端子の接続関係を指示する仮引数部603と、二分木ゲートテーブルの端子を表す端子テーブル604と、二分木ゲートテーブルのゲートを表すゲートテーブル605から構成される。ひな型ゲート展開文508についても同様である。

第7図は論理マクロデータ412からひな型展

し、その入出端子をZ, c(0), m2に変換する(701)。制御文507は、'('と')'の間の4回繰返し1回毎にiを0から3までインクリメントする。ひな型ゲートテーブル展開文508は制御文507によって4回ひな型ゲートテーブル ' halfsub ' を複写し入出力端子を変換する(702)。制御文509を変数iにn-1をセットする。制御関数文510はm1(0-i)すなわちm1(0), m1(1), m1(2), m1(3)の4ビットを入力端子としm2を出力端子とするorゲートテーブルを生成する(703)。同一名の入出力端子どうしを接続して二分木ゲートテーブルを生成する。

第8図は上記例により生成された二分木ゲートテーブルである。

ステップ104: 本ステップは二分木ゲートテーブル800を入力しゲートレベル論理ファイル113を出力する。

第9図は、二分木ゲートテーブル800から生成されるゲートのゲート図900である。

開ルールデータ500とひな型ゲートテーブル600, 601を用いて二分木ゲートテーブルを生成する過程を表したものである。以下では、第5, 第6, 第7図を用いて本発明による論理マクロ展開の処理例を説明する。

ステップ100で抽出された論理マクロ700の論理マクロ名 ' \leq ' (比較演算)によりひな型展開ルールデータ500が検索され引数として仮引数502n, a, b, zに4, A, B, Zが渡される。仮引数宣言部503によりn=4はビット幅を表しa=A, b=Bは4ビットz=Zは1ビットのパラメタであることを表す。内部宣言部504ではcが4+1ビット, m1が4ビット, m2が1ビットのひな型展開ルール内部の接続関係を表す内部パラメタであることを表し、iが制御文で使用する内部変数であることを表す。展開ルール記述部505は、まずひな型ゲートテーブル展開文506でひな型ゲートテーブル名 ' \leq head' をキーにひな型ゲート定義データテーブル112を検索し当該ひな型ゲートテーブルを複写

本実施例によれば、演算式から所望のゲート論理をビット幅を意識することなく生成することが可能である。

【発明の効果】

本発明によれば、最大ビット幅の上限がない論理マクロ展開方式を提供することができるので、演算子論理展開においてビット幅の上限を意識することなく設計することができるという効果がある。

また、ひな型展開ルール、ひな型ゲートテーブルは展開すべき論理機能のビット幅に依存しないので、必要とされる論理機能のビット幅の増加によるメンテナンスが不要であり、その工数を削減できる効果もある。

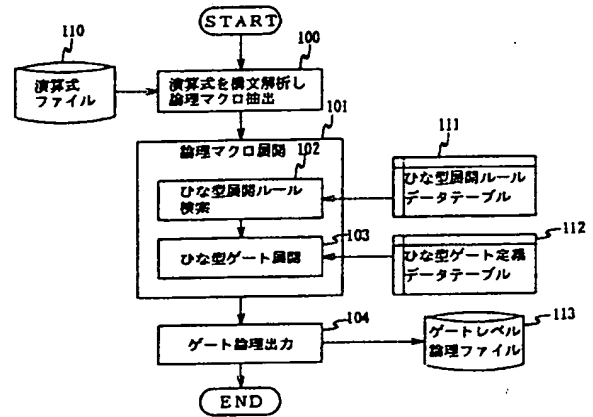
4. 図面の簡単な説明

第1図は本発明の一実施例としての演算式論理マクロ展開の概略処理フロー、第2図は本発明の一実施例としてのゲート論理生成の概略処理フロー、第3図は本発明の一実施例としての論理動作記述図、第4図は本発明の一実施例としての演算

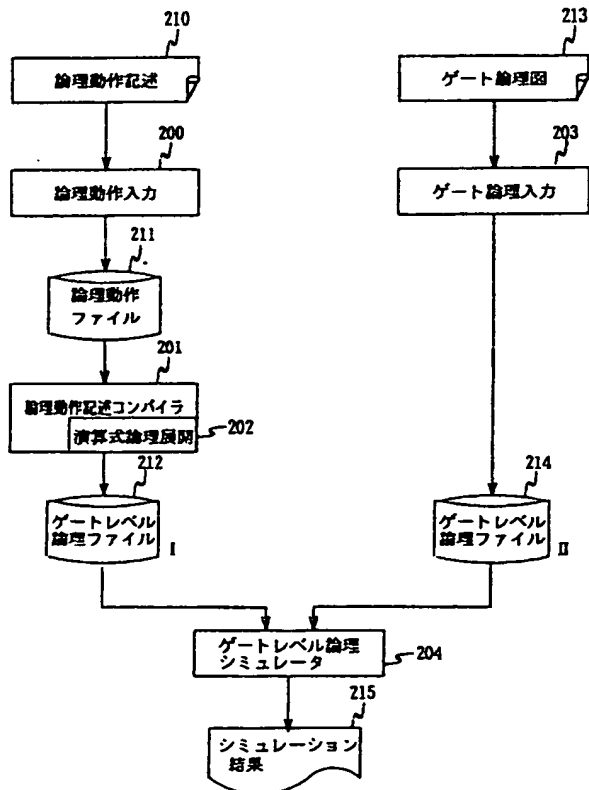
式、論理マクロ抽出の処理フロー、第5図は本発明の一実施例としてのひな型展開ルール、第6図は本発明の一実施例としてのひな型ゲートテーブル、第7図は本発明の一実施例としての論理マクロ展開図、第8図は本発明の一実施例としての論理マクロから生成される二分木ゲートテーブル、第9図は本発明の一実施例としての二分木ゲートテーブルから生成されるゲートのゲート図を示す。
 110…演算式ファイル、101…論理マクロ展開、102…ひな型展開ルール検索、103…ひな型ゲート展開、111…ひな型展開ルールデータテーブル、112…ひな型ゲート定義データテーブル、113…ゲートレベル論理ファイル。

代理人 井理士 小川勝男

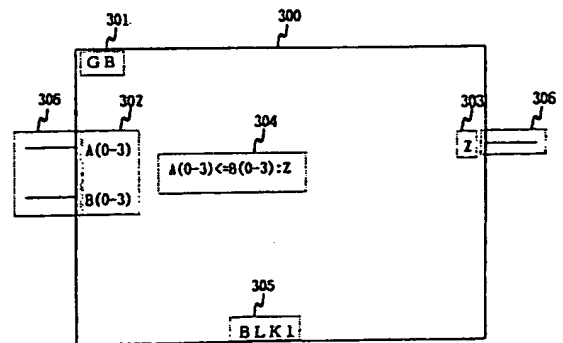
第1図



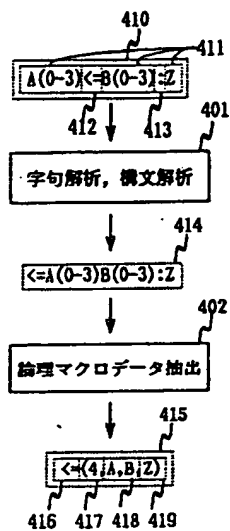
第2図



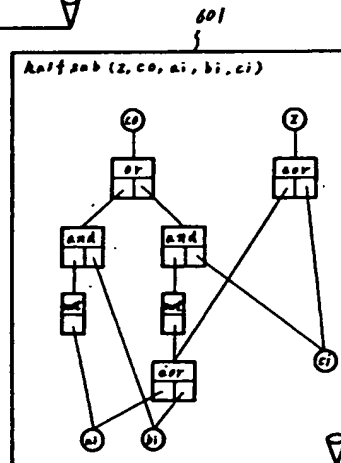
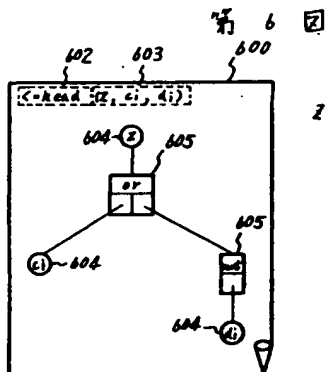
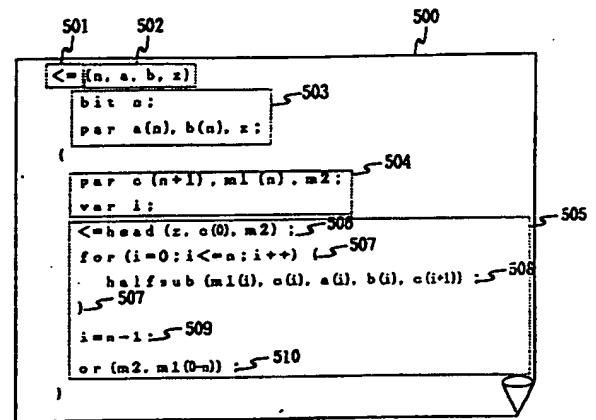
第3図



第4図



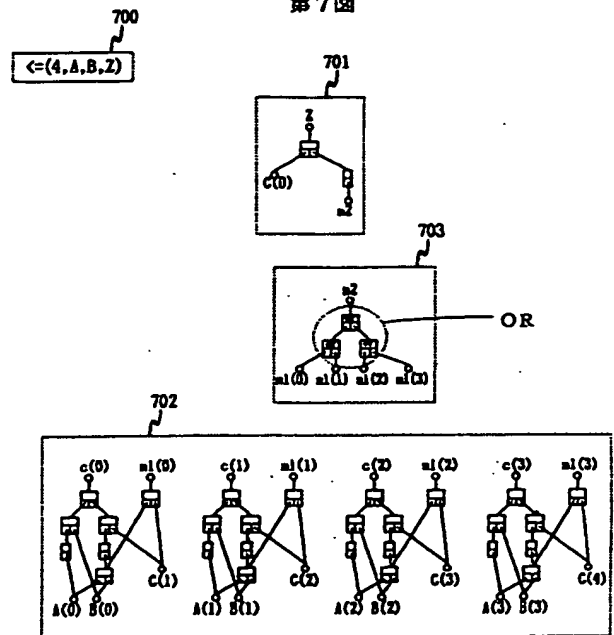
第5図



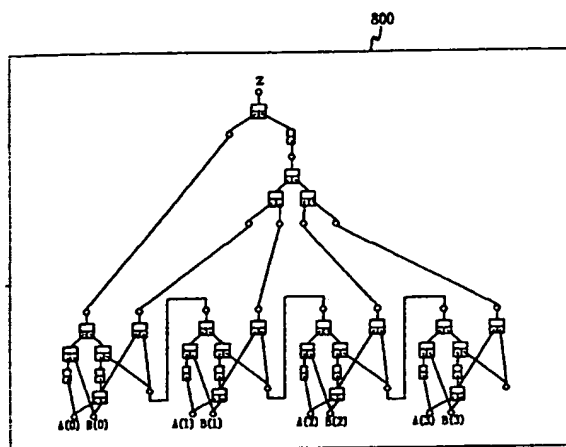
$$c = (A_i \oplus b_i) \oplus c_i$$

$$z = (A_i \oplus b_i) \oplus c_i$$

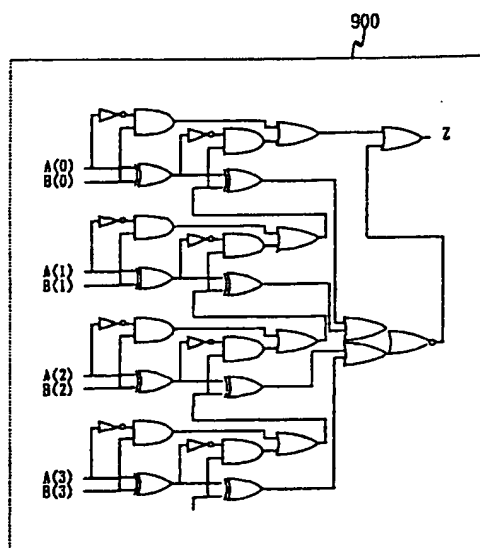
第7図



第8図



第9図



第1頁の続き

⑩発 明 者 天 野 亘 孝

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

⑪発 明 者 本 村 哲 朗

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

⑫発 明 者 福 永 明

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

⑬発 明 者 西 坂 利 夫

神奈川県秦野市堀山下1番地 株式会社日立製作所神奈川工場内